Initial Process Security

Damas P. GRUSKA[1] and M. Carmen Ruiz[2]

[1] Comenius University, Slovakia
[2] Universidad de Castilla-La Mancha, Spain

Abstract. A new security property called *initial process opacity* is formalized and studied. It assumes an intruder who partially observes system's execution(s) and tries to deduce information on its initial state(s) , which is expressed by a given predicate. Systems are considered to be secure, if this is not possible. The property and its variants are investigated and compared. As a basic formalism timed process algebra is exploited.

Keywords: security, opacity, process algebras, information flow

1 Introduction

The internet of things (IoT) brings new challenges. Each device connected increases privacy and security concerns surrounding the IoT. These concerns range from hackers stealing our data and even threatening our lives to how corporations can easily uncover private data we carelessly give them. The great revolution brought about by IoT involves the emergence of new devices, new protocols and, of course, new security needs to fulfill the new requirements. New protocols come into operation before they have been evaluated in depth. Which leads to the appearance of new versions of the protocol that is not always compatible with its predecessors and that companies will not always incorporate in their devices with sufficient speed. In addition, these solutions usually require downloading new code and this itself is open to security attacks.

An example of this can be found in [Gar16] where we present an architecture for Wireless Sensor and Actuator Networks (WSAN) using the Bluetooth Low Energy (BLE) and TCP/IP protocols in conjunction, which make necessary to include bridges that lack basic security requirements. Another example is found in [Hor17] where we propose a new packet format and a new BLE mesh topology, with two different configurations: Individual Mesh and Collaborative Mesh. To include user devices is a challenge because of the minimal capacity devices being used, the physical accessibility to sensors, actuators and objects, and the openness of the systems, including the fact that most devices will communicate wirelessly.

Tests and simulations are the usual validation techniques used for algorithms and protocols but we advocate the use of formal methods for the evaluation of such algorithms due several reasons: simulation results depend on the simulator and vary with those obtained in the field experiments. Moreover, testing and simulation can only show the presence of errors, not their absence but to rule out errors we must consider all possible executions. This can be made by means of formal methods that provides correct results that cover the full behaviour of the models. At present there are few works where formal methods are used in IoT and they are focused on the field of automotive.

Here we focus on formal methods applied in a field of security of IoT protocols. Communications aspects of these protocols are consider to be the weakest point. Many new, general or proprietary protocols are frequently vulnerable by various attacks. Here we propose, by means of formal models and formal methods, a way how to define and how to check some aspects of security of IoT, namely low level protocols. We exploit information flow based security properties (see [GM82]) which assume an absence of any information flow between private and public systems activities. This means that systems are considered to be secure if from observations of their public activities no information about private activities or states can be deduced. This approach has found many reformulations and among them opacity (see [BKR04,BKMR06]) could be considered as the most general one and many other security properties could be viewed as its special cases (see, for example, [Gru07]).

Opacity properties could be divided into two types: language based opacity, expressing security of system's actions and state based one, concentrating on system's states (see an overview paper [JLF16]). The former one is much more studied for process algebra's formalism. But also for the later one there is some research already done. In [Gru15] opacity is modified (the result is called process opacity) in such a way that instead of process's traces we focus on properties of reachable states. Hence it assumes an intruder who is not primarily interested whether some sequence of actions performed by a given process has some given property. Instead of that we consider an intruder who wants to discover whether a process reaches a state which always satisfies a given predicate. It turns out that in this way some new security flaws could be expressed.

In this paper we assume an intruder who observes system's execution and tries to deduce information on its initial state. Systems are considered to be secure, if this is not possible. This security property will be called initial process opacity according to initial state opacity for discrete event systems (see [JLF16]). It assumes that an intruder cannot see all actions but only visible ones, and tries to deduce whether an initial state of computation satisfies some given predicate ϕ. This is not possible, if there exists another initial state, for which ϕ does not hold but from that state the same visible computation, i.e. a sequence of visible actions, could be performed. We define also two variants of this property, namely persistent and strong initial process opacity. We prove some basic properties for these security properties. Particularly we investigate finite state systems and time sensitive observations.

Since our plan is to elaborate techniques for description of timing attacks and to verify systems security against them, we have decided to work with a timed process algebra which can be used for description of timing behavior of systems. We do not consider value-passing algebra since we focus on actions and not on communicated values. Considering also values and possible security types of variables would bring new challenges and we leave it for future work together with some other proposals which are mentioned in the next sections.

The paper is organized as follows. In Section 2 we describe the timed process algebra TPA which will be used as a basic formalism. In Section 3 we present some notion on information flow security and in the next section initial process opacity and its variants are defined, studied and compared. Section 5 contains discussion and plans for future work.

2 Timed Process Algebra

In this section we define Timed Process Algebra, TPA for short. TPA is based on Milner's CCS but the special time action t which expresses elapsing of (discrete) time is added. The presented language is a slight simplification of Timed Security Process Algebra introduced in [FGM00]. We omit an explicit idling operator ι used in tSPA and instead of this we allow implicit idling of processes. Hence processes can perform either "enforced idling" by performing t actions which are explicitly expressed in their descriptions or "voluntary idling" (i.e. for example, the process $a.Nil$ can perform t action since it is not contained the process specification). But in both cases internal communications have priority to action t in the parallel composition. Moreover we do not divide actions into private and public ones as it is in tSPA. TPA differs also from the tCryptoSPA (see [GM04]). TPA does not use value passing and strictly preserves *time determinancy* in case of choice operator $+$ what is not the case of tCryptoSPA.

To define the language TPA, we first assume a set of atomic action symbols A not containing symbols τ and t, and such that for every $a \in A$ there exists $\overline{a} \in A$ and $\overline{\overline{a}} = a$. We define $Act = A \cup \{\tau\}, At = A \cup \{t\}, Actt = Act \cup \{t\}$. We assume that $a, b, \dots$ range over A, $u, v, \dots$ range over Act, and $x, y \dots$ range over $Actt$. Assume the signature $\Sigma = \bigcup_{n \in \{0,1,2\}} \Sigma_n$, where

$$\begin{aligned}
\Sigma_0 &= \{Nil\} \\
\Sigma_1 &= \{x. \mid x \in A \cup \{t\}\} \cup \{[S] \mid S \text{ is a relabeling function}\} \\
&\quad \cup \{\backslash M \mid M \subseteq A\} \\
\Sigma_2 &= \{|, +\}
\end{aligned}$$

with the agreement to write unary action operators in prefix form, the unary operators $[S], \backslash M$ in postfix form, and the rest of operators in infix form. Relabeling functions, $S : Actt \to Actt$ are such that $\overline{S(a)} = S(\bar{a})$ for $a \in A, S(\tau) = \tau$ and $S(t) = t$.

The set of TPA terms over the signature Σ is defined by the following BNF notation:

$$P \ ::= \ X \ \mid \ op(P_1, P_2, \dots P_n) \ \mid \ \mu X P$$

where $X \in Var$, Var is a set of process variables, $P, P_1, \dots P_n$ are TPA terms, $\mu X-$ is the binding construct, $op \in \Sigma$.

The set of CCS terms consists of TPA terms without t action. We will use an usual definition of opened and closed terms where μX is the only binding operator. Closed terms which are t-guarded (each occurrence of X is within some subterm $t.A$ i.e. between any two t actions only finitely many non timed actions can be performed) are called TPA processes.

We give a structural operational semantics of terms by means of labeled transition systems. The set of terms represents a set of states, labels are actions from $Actt$. The transition relation $\rightarrow$ is a subset of TPA $\times$ $Actt \times$ TPA. We write $P \overset{x}{\rightarrow} P'$ instead of $(P, x, P') \in \rightarrow$ and $P \overset{x}{\not\rightarrow}$ if there is no P' such that $P \overset{x}{\rightarrow} P'$. The meaning of the expression $P \overset{x}{\rightarrow} P'$ is that the term P can evolve to P' by performing action x, by $P \overset{x}{\rightarrow}$ we will denote that there exists a term P' such that $P \overset{x}{\rightarrow} P'$. We define the transition relation as the least relation satisfying the inference rules for CCS plus the following inference rules:

$$\frac{}{Nil \overset{t}{\rightarrow} Nil}\ A1 \qquad\qquad \frac{}{u.P \overset{t}{\rightarrow} u.P}\ A2$$

$$\frac{P \overset{t}{\rightarrow} P', Q \overset{t}{\rightarrow} Q', P \mid Q \overset{\tau}{\not\rightarrow}}{P \mid Q \overset{t}{\rightarrow} P' \mid Q'}\ Pa \qquad \frac{P \overset{t}{\rightarrow} P', Q \overset{t}{\rightarrow} Q'}{P + Q \overset{t}{\rightarrow} P' + Q'}\ S$$

Here we mention the rules that are new with respect to CCS. Axioms $A1, A2$ allow arbitrary idling. Concurrent processes can idle only if there is no possibility of an internal communication (Pa). A run of time is deterministic (S) i.e. performing of t action does not lead to the choice between summands of $+$. In the definition of the labeled transition system we have used negative premises (see Pa). In general this may lead to problems, for example with consistency of the defined system. We avoid these dangers by making derivations of τ independent of derivations of t. For an explanation and details see [Gro90].

For $s = x_1.x_2.\ldots.x_n, x_i \in Actt$ we write $P \overset{s}{\rightarrow}$ instead of $P \overset{x_1}{\rightarrow}\overset{x_2}{\rightarrow} \ldots \overset{x_n}{\rightarrow}$ and we say that s is a trace of P. The set of all traces of P will be denoted by $Tr(P)$. By ϵ we will denote the empty sequence of actions, by $Succ(P)$ we will denote the set of all successors of P i.e. $Succ(P) = \{P' | P \overset{s}{\rightarrow} P', s \in Actt^*\}$. If the set $Succ(P)$ is finite we say that P is a finite state process. We define modified transitions $\overset{x}{\Rightarrow}_M$ which "hide" actions from M. Formally, we will write $P \overset{x}{\Rightarrow}_M P'$ for $M \subseteq Actt$ iff $P \overset{s_1}{\rightarrow}\overset{x}{\rightarrow}\overset{s_2}{\rightarrow} P'$ for $s_1, s_2 \in M^\star$ and $P \overset{s}{\Rightarrow}_M$ instead of $P \overset{x_1}{\Rightarrow}_M \overset{x_2}{\Rightarrow}_M \ldots \overset{x_n}{\Rightarrow}_M$. We will write $P \overset{x}{\Rightarrow}_M$ if there exists P' such that $P \overset{x}{\Rightarrow}_M P'$. We will write $P \widehat{\overset{x}{\Rightarrow}}_M P'$ instead of $P \overset{\epsilon}{\Rightarrow}_M P'$ if $x \in M$. Note that $\overset{x}{\Rightarrow}_M$ is defined for arbitrary action but in definitions of security properties we will use it for actions (or sequence of actions) not belonging to M. We can extend the definition of $\Rightarrow_M$ for sequences of actions similarly to $\overset{s}{\rightarrow}$.

Let $s \in Actt^\star$. By $|s|$ we will denote the length of s i.e. a number of action contained in s. By $s|_B$ we will denote the sequence obtained from s by removing all actions not belonging to B. For example, $|s|_{\{t\}}|$ denote a number of occurrences of t in s, i.e. time length of s.

By $Sort(P)$ we will denote the set of actions from A which can be performed by P. The set of weak timed traces of process P is defined as $Tr_w(P) = \{s \in (A \cup \{t\})^\star | \exists P'.P \overset{s}{\Rightarrow}_{\{\tau\}} P'\}$. Two processes P and Q are weakly timed trace equivalent ($P \simeq_w Q$) iff $Tr_w(P) = Tr_w(Q)$. We conclude this section with definitions of M-bisimulation and weak timed trace equivalence.

Definition 1. *Let $(TPA, Actt, \rightarrow)$ be a labelled transition system (LTS). A relation $\Re \subseteq TPA \times TPA$ is called a* M-bisimulation *if it is symmetric and it satisfies the following condition: if $(P, Q) \in \Re$ and $P \overset{x}{\rightarrow} P', x \in Actt$ then there exists a process Q' such that $Q \overset{\widehat{x}}{\Rightarrow}_M Q'$ and $(P', Q') \in \Re$. Two processes P, Q are M-bisimilar, abbreviated $P \approx_M Q$, if there exists a M-bisimulation relating P and Q.*

3 Information flow

In this section we will present motivations for new security concepts which will be introduced in the next section. First we define the absence-of-information-flow property - Strong Nondeterministic Non-Interference (SNNI, for short, see [FGM00]). Suppose that all actions are divided into two groups, namely public (low level) actions L and private (high level) actions H. It is assumed that $L \cup H = A$. SNNI property assumes an intruder who tries to learn whether a private action was performed by a given process while (s)he can observe only public ones. If this cannot be done then the process has SNNI property. Formally, process P has SNNI property (we will write $P \in SNNI$) if $P \setminus H$ behaves like P for which all high level actions are hidden (namely, replaced by action τ) for an observer. To express this hiding we introduce the hiding operator $P/M, M \subseteq A$, for which it holds that if $P \overset{a}{\rightarrow} P'$ then $P/M \overset{a}{\rightarrow} P'/M$ whenever $a \notin M \cup \bar{M}$ and $P/M \overset{\tau}{\rightarrow} P'/M$ whenever $a \in M \cup \bar{M}$. We say that P has SNNI property, and we write $P \in SNNI$ iff $P \setminus H \simeq_w P/H$. A generalization of this concept is given by opacity (this concept was exploited in [BKR04], [BKMR06] and [Gru07] in a framework of Petri Nets, transition systems and process algebras, respectively). Actions are not divided into public and private ones at the system description level but a more general concept of observations and predicates are exploited. A predicate is opaque if for any trace of a system for which it holds, there exists another trace for which it does not hold and the both traces are indistinguishable for an observer (which is expressed by an observation function). This means that the observer (intruder) cannot say whether a trace for which the predicate holds has been performed or not.

Let us assume that an intruder tries to discover whether a given process can reach a state with some given property which is expressed by a (total) predicate. This might be process deadlock, capability to execute only traces with time length less then n time unites, capability to perform at the same time actions form a given set, incapacity to idle (to perform t action) etc. We do not put any restriction on such predicates but we only assume that they are consistent with some suitable behaviorial equivalence. The formal definition follows.

Definition 2. *We say that the predicate ϕ over processes is consistent with respect to relation $\cong$ if whenever $P \cong P'$ then $\phi(P) \Leftrightarrow \phi(P')$.*

As the consistency relation $\cong$ we could take bisimulation ($\approx_{\emptyset}$), weak bisimulation ($\approx_{\{\tau\}}$) or any other suitable equivalence. A special class of such predicates are such ones (denoted as $\phi_{\cong}^{Q}$) which are defined by a given process Q and equivalence relation $\cong$ i.e. $\phi_{\cong}^{Q}(P)$ holds iff $P \cong Q$.

We suppose that the intruder can observe only some activities performed by the process. Hence we suppose that there is a set of public actions which can be observed and a set of hidden (not necessarily private) actions. To model such observations we exploit the relation $\overset{s}{\Rightarrow}_M$ where actions from M are those ones which could not be seen by the observer. The formal definition of process opacity (see [Gru15]) is the following.

Definition 3 (Process Opacity). *Given process P, a predicate ϕ over processes is process opaque w.r.t. the set M if whenever $P \overset{s}{\Rightarrow}_M P'$ for $s \in (Actt \setminus M)^*$ and $\phi(P')$ holds then there exists P'' such that $P \overset{s}{\Rightarrow}_M P''$ and $\neg\phi(P'')$ holds. The set of processes for which the predicate ϕ is process opaque w.r.t. to the M will be denoted by POp_M^{ϕ}.*

Note that if $P \cong P'$ then $P \in POp_M^{\phi} \Leftrightarrow P' \in POp_M^{\phi}$ whenever ϕ is consistent with respect to $\cong$ and $\cong$ is such that it is a subset of the trace equivalence (defined as $\simeq_w$ but instead of $\overset{s}{\Rightarrow}_{\{\tau\}}$ we use $\overset{s}{\Rightarrow}_{\emptyset}$).

$$P \quad \overset{s}{\Longrightarrow}_M \quad \phi(P')$$

$$P \quad \overset{s}{\Longrightarrow}_M \quad \neg\phi(P'')$$

Fig. 1. Process opacity

4 Initial Process Opacity

In this section we will define and study a new security property called initial process opacity. It is, in a sense, the opposite of process opacity. While the former looks backwards, the later looks forward. Initial process opacity formalizes an intruder who tries to learn an initial state of a system seeing only the visible part of the execution leading to the current system's state. For discrete event systems a formalization of initial process opacity can be found in an overview paper [JLF16] where it is called initial state opacity. To define initial process opacity for process algebras we assume a predicate ϕ over processes and a set of initial states $\mathcal{P}$. Moreover, we assume that an intruder cannot see actions contained in $M, M \subseteq Actt$. Initial process opacity requires that if it is possible to reach the current state (process) P from an initial state which satisfies ϕ there should exist another initial state from $\mathcal{P}$ which does not satisfy ϕ but from which the observably same execution path leads to a state indistinguishable (by $\approx_M$) from P. The formal definition is the following.

Definition 4 (Initial Process Opacity). *Given process P and a set of initial processes $\mathcal{P}$. A predicate ϕ over processes is initially process opaque w.r.t. the sets M and $\mathcal{P}$ if whenever $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}$ and $\phi(P')$ holds then there exists Q, Q', $Q' \in \mathcal{P}$ such that $\neg\phi(Q')$ holds, $Q' \overset{s}{\Rightarrow}_M Q$ and $P \approx_M Q$.*

The set of processes for which the predicate ϕ is initially process opaque w.r.t. to the sets $\mathcal{P}$ and M will be denoted by $IPOp_M^{\phi}(\mathcal{P})$.

Differences between process opacity and initial process opacity are depicted on Fig 1 and 2.

$$\begin{tikzcd}
\phi(P'), P' \arrow[r, Rightarrow, "s", "M"'] & P \arrow[d, phantom, "\approx_M"] \\
\neg\phi(Q'), Q' \arrow[r, Rightarrow, "s", "M"'] & Q
\end{tikzcd}$$

Fig. 2. Initial process opacity

4.1 Properties

In this subsection we present some properties of initial process opacity. We start with inclusion propositions, which express how sets $\mathcal{P}, M$ and power of predicate ϕ influence the set $IPOp_M^{\phi}(\mathcal{P})$. We define a predicate ϕ also for sets of processes in the following way. Let T be a set of processes, then $\phi(T)$ holds if it holds for every process from T.

Proposition 1. *Let $\mathcal{P}_1 \subseteq \mathcal{P}_2$ and $\neg\phi(\mathcal{P}_2 \setminus \mathcal{P}_1)$ holds. Then $IPOp_M^{\phi}(\mathcal{P}_1) \subseteq IPOp_M^{\phi}(\mathcal{P}_2)$.*

Proof. Let $P \in IPOp_M^{\phi}(\mathcal{P}_1)$. We will show that also $P \in IPOp_M^{\phi}(\mathcal{P}_2)$. Let there exists an initial state P' from $\mathcal{P}_2$ such that $\phi(P')$ holds and $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$. By the assumption we know that $P' \in \mathcal{P}_1$ and since $P \in IPOp_M^{\phi}(\mathcal{P}_1)$ we know that there exists $Q, Q \in \mathcal{P}_1$ which fulfills Definition 4. But since also $Q \in \mathcal{P}_2$ we have $P \in IPOp_M^{\phi}(\mathcal{P}_2)$.

Note that in general the set inclusion in the previous proposition cannot be replaced by the equation as well as the assumption that the two sets of initial states could differ only by processes for which ϕ does not hold, cannot be omitted.

Proposition 2. *Let $\phi_1 \Rightarrow \phi_2$. Then $IPOp_M^{\phi_2}(\mathcal{P}) \subseteq IPOp_M^{\phi_1}(\mathcal{P})$.*

Proof. Let $P \in IPO_M^{\phi_2}(\mathcal{P})$ we will show that also $P \in IPOp_M^{\phi_1}(\mathcal{P})$. Let $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}$ and $\phi_1(P')$ holds. Then since $\phi_1 \Rightarrow \phi_2$ we have that also $\phi_2(P')$ holds. Since $P \in IPO_M^{\phi_2}(\mathcal{P})$ there exists Q, Q', $Q' \in \mathcal{P}$ such that $\neg\phi_2(Q')$ holds, $Q' \overset{s}{\Rightarrow}_M Q$ and $P \approx_M Q$. Again since $\neg\phi_2 \Rightarrow \neg\phi_1$ we know that $\neg\phi_1(Q')$ holds what means that $P \in IPOp_M^{\phi_1}(\mathcal{P})$.

Proposition 3. *Let $M_1 \subseteq M_2$. Then $IPOp_{M_1}^{\phi}(\mathcal{P}) \subseteq IPOp_{M_2}^{\phi}(\mathcal{P})$.*

Proof. Let $P \in IPO_{M_1}^{\phi}$ we will show that also $P \in IPOp_{M_2}^{\phi}(\mathcal{P})$. Let $P' \overset{s}{\Rightarrow}_{M_1} P$ for $s \in (Actt \setminus M_2)^*$, $P' \in \mathcal{P}$ and $\phi(P')$ holds. Since $M_1 \subseteq M_2$ we have $s \in (Actt \setminus M_1)^*$ and hence then there exists Q, Q', $Q' \in \mathcal{P}$ such that $\neg\phi(Q')$ holds, $Q' \overset{s}{\Rightarrow}_{M_1} Q$ and $P \approx_{M_1} Q$. Since it is easy to show that $\approx_{M_1} \subseteq \approx_{M_2}$ what means that $P \in IPOp_{M_2}^{\phi}(\mathcal{P})$.

Proposition 4. *Let $P \in IPOp_M^{\phi}(\mathcal{P}_1)$ and $P \in IPOp_M^{\phi}(\mathcal{P}_2)$. Then $P \in IPOp_M^{\phi}(\mathcal{P}_1 \cup \mathcal{P}_2)$.*

Proof. Let $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}_1 \cup \mathcal{P}_2$ and $\phi(P')$ holds. Without loss of generality we can assume that $P' \in \mathcal{P}_1$. Since $P \in IPOp_M^{\phi}(\mathcal{P}_1)$ there exists Q, Q', $Q' \in \mathcal{P}_1$ such that $\neg\phi(Q')$ holds, $Q' \overset{s}{\Rightarrow}_M Q$ and $P \approx_M Q$. Since $Q' \in \mathcal{P}_1 \cup \mathcal{P}_2$ we have $P \in IPOp_M^{\phi}(\mathcal{P}_1 \cup \mathcal{P}_2)$.

Note that a similar proposition does not hold for the set intersection, i.e. $P \in IPOp_M^{\phi}(\mathcal{P}_1)$ and $P \in IPOp_M^{\phi}(\mathcal{P}_2)$ does not imply that $P \in IPOp_M^{\phi}(\mathcal{P}_1 \cap \mathcal{P}_2)$.

Under some special assumptions all processes belong to $IPOp_M^{\phi}(\mathcal{P})$. Note, that for those ones which are not successors of any process from $\mathcal{P}$ this holds directly from Definition 4.

Proposition 5. *Let for a given predicate ϕ and set M every equivalence class of $TPA/\approx_M$ contains processes P, Q such that $\phi(P)$, and $\neg\phi(Q)$. Then $TPA = IPOp_M^{\phi}(\mathcal{P})$.*

Proof. Sketch. Let $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}$ and $\phi(P')$ holds. According to the assumption there exists Q, $Q' \approx_M P'$ such that $\neg\phi(Q')$ holds. Moreover, we have $Q' \overset{s}{\Rightarrow}_M Q$ and $P \approx_M Q$. Hence $P \in IPOp_M^{\phi}(\mathcal{P})$. If there does not exist P' such that $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$ for some $P' \in \mathcal{P}$ such that $\phi(P')$ holds, then automatically $P \in IPOp_M^{\phi}(\mathcal{P})$.

Initial process opacity could be inherited under some special assumptions as it is stated by the following property.

Proposition 6. *Assume that whenever $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}$ and $\phi(P')$ holds then there exist $s_1, s_2 \in (Actt \setminus M)^*$ such that $s = s_1.s_2$ and R such that $P' \overset{s_1}{\Rightarrow}_M R \overset{s_2}{\Rightarrow}_M P$ and $R \in IPOp_M^{\phi}(\mathcal{P})$. Then $P \in IPOp_M^{\phi}(\mathcal{P})$.*

Proof. Sketch. Let $P' \overset{s}{\Rightarrow}_M P$ for $s \in (Actt \setminus M)^*$, $P' \in \mathcal{P}$ and $\phi(P')$ holds. Then there exist $s_1, s_2 \in (Actt \setminus M)^*$ such that $s = s_1.s_2$ and R such that $P' \overset{s_1}{\Rightarrow}_M R \overset{s_2}{\Rightarrow}_M P$ and $R \in IPOp^{\phi}_M(\mathcal{P})$. Since $R \in IPOp^{\phi}_M(\mathcal{P})$ there exists Q, Q', $Q' \in \mathcal{P}$ such that $\neg\phi(Q')$ holds, $Q' \overset{s_1}{\Rightarrow}_M Q$ and $R \approx_M Q$. But then there exists process Q_1 such that $Q \overset{s_2}{\Rightarrow}_M Q_1$ and $P \approx_M Q_1$ and hence $P \in IPOp^{\phi}_M(\mathcal{P})$.

Persistent variants of security properties requires that if system's state is secure then all its subsequent states are secure as well. Now we will formally define a persistent variant of initial process opacity.

Definition 5 (Persistent Initial Process Opacity). *We say that process P is persistently initial state opaque with respect to $M, \phi, \mathcal{P}$ (denoted $P \in PIPOp^{\phi}_M(\mathcal{P})$) if $P \in IPOp^{\phi}_M(\mathcal{P})$ and whenever $P' \in Succ(P)$ then $P' \in IPOp^{\phi}_M(\mathcal{P})$.*

Persistent initial process opacity is the stronger property than initial process opacity as it is stated by the following proposition.

Proposition 7. *$PIPOp^{\phi}_M(\mathcal{P}) \subseteq IPOp^{\phi}_M(\mathcal{P})$ moreover there exist $M, \phi, \mathcal{P}$ such that $PIPOp^{\phi}_M(\mathcal{P}) \subset IPOp^{\phi}_M(\mathcal{P})$.*

Proof. Sketch. The first part follows directly from Definition 5. A hint for the second part comes from the proof of Proposition 3 which holds only under some assumptions, i.e. not for all successors of initially state opaque process are initially state opaque as well.

Initial process opacity guarantees that an intruder cannot learn whether an initial state satisfy given predicate ϕ. But it is not useful if also validity of $\neg\phi$ is interesting for intruders. Hence we define strong initial process opacity.

Definition 6 (Strong Initial Process Opacity). *We say that process P is strongly initial state opaque with respect to $M, \phi, \mathcal{P}$ (denoted $P \in SIPOp^{\phi}_M(\mathcal{P})$) if $P \in IPOp^{\phi}_M(\mathcal{P})$ and $P \in IPOp^{\neg\phi}_M(\mathcal{P})$.*

Clearly, strong initial process opacity is stronger than initial process opacity as it is stated by the following Lemma.

Lemma 1. *$SIPOp^{\phi}_M(\mathcal{P}) \subseteq IPOp^{\phi}_M(\mathcal{P})$ moreover there exist $M, \phi, \mathcal{P}$ such that $SIPOp^{\phi}_M(\mathcal{P}) \subset IPOp^{\phi}_M(\mathcal{P})$.*

Proof. Sketch. The first part follows directly from Definition 6. A hint for the second part comes from the proof of Proposition 5 which holds only under some assumptions, i.e. that every equivalence class of $\approx_M$ contains processes P and Q for which $\phi(P)$ and $\neg\phi(Q)$ hold, respectively.

4.2 Finite state systems

From now on we will consider finite state processes (systems) i.e. processes for which $Succ(P)$ is a finite set. Typically such systems could be communication protocols, hardware components etc. Let us consider a set of processes $\mathcal{T}$. By $Succ(\mathcal{T})$ we will denote the set of all successors of processes from $\mathcal{T}$, formally

$$Succ(\mathcal{T}) = \bigcup_{P \in \mathcal{T}} Succ(P).$$

Now we can define secure systems, i.e. systems for which every state is secure with respect to $IPOp_M^{\phi}(\mathcal{P})$ property.

Definition 7 (Secure System). *We say that the set of processes $\mathcal{S}$ is secure with respect to $IPOp_M^{\phi}(\mathcal{P})$ property iff $\mathcal{S} \subseteq IPOp_M^{\phi}(\mathcal{P})$.*

Proposition 8. *Let for given predicate ϕ every equivalence class of $\mathcal{P}/\approx_M$ contains processes P, Q such that $\phi(P)$, and $\neg\phi(Q)$. Then $IPOp_M^{\phi}(\mathcal{P}) = Succ(\mathcal{P})$.*

Proof. Directly from Proposition 5.

In general, systems security with respect to $IPOp_M^{\phi}(\mathcal{P})$ property is undecidable. But a significant class of systems is secure as it is stated by the following proposition.

Proposition 9. *Let given predicates $\phi, \neg\phi$ are decidable predicates, $\mathcal{P}$ is a finite set of finite state processes. Then security of any set $\mathcal{T}, \mathcal{T} \subseteq TPA$ with respect to $IPOp_M^{\phi}(\mathcal{P})$ is a decidable property.*

Proof. The main idea. It is easy to see that it is enough to prove that security of $Succ(\mathcal{P})$ is decidable, since for every $P \in T \setminus Succ(\mathcal{P})$ we have $P \in IPOp_M^{\phi}(\mathcal{P})$. Decidability of security of $Succ(\mathcal{P})$ follows from the fact that it is finite and we can simply "generate" all successors for processes from $\mathcal{P}$ check whether ϕ or $\neg\phi$ holds for them (according to the assumption, predicates $\phi, \neg\phi$ are decidable). Note that to prove the proposition we cannot use Proposition 8 since its reverse does not hold.

4.3 Time sensitive observations

Time attacks belong to powerful tools for attackers who can observe or interfere with systems in real time. On the other side this techniques is useless for off line systems and hence they could be consider safe with respect to attackers who cannot observe (real) time. By the presented formalism we have a way how to distinguish these two cases.

Definition 8 (Immunity with respect to Timinig Attacks). *We say that process P is immune to timing attacks with respect to ϕ and $M, t \notin M$ iff $P \notin IPOp_M^{\phi}(\mathcal{P})$ but $P \in IPOp_{\{M \cup t\}}^{\phi}(\mathcal{P})$.*

Moreover, we can define time after which the systems are secure even with respect to timing attacks.

Definition 9 (n-Secure Systems). *System $Succ(\mathcal{P})$ is called n-secure with respect to $IPOp_M^{\phi}(\mathcal{P})$ property whenever $P \overset{s}{\Rightarrow}_M Q$ for $P \in \mathcal{P}$ and $s|_{\{t\}} \geq n$ then $Q \in IPOp_M^{\phi}(\mathcal{P})$.*

For n-secure system $Succ(\mathcal{P})$ we know that after elapsing n time unites all states are secure with respect to $IPOp_M^{\phi}(\mathcal{P})$ property. Moreover, n-security is a decidable property as it is stated by the following proposition.

Proposition 10. *Let given predicates $\phi, \neg\phi$ are decidable, $\mathcal{P}$ is the finite set of finite state processes. Then n-security of $Succ(\mathcal{P})$ with respect to $IPOp_M^{\phi}(\mathcal{P})$ is a decidable property.*

Proof. The main idea. We exploit decidability of $IPOp_M^{\phi}(\mathcal{P})$ property (Proposition 9).

Note that n-security is a property which could be found in systems which exhibit "cyclic" behaviour which does not include initial states, i.e. after initialization they enter in some cycle whose all states are secure with respect to $IPOp_M^{\phi}(\mathcal{P})$ property. Systems which periodically enter some of initial states could exhibit just "secure time windows" or phases, in which they are secure but after some computation they could enter into states which are not secure anymore. We leave investigation of such systems to future work.

To overcome these kinds of security flaws, usually a random delays technique is employed, which add some random time delays in executions. The presented research can help us to spot precise places where such delays are needed and to count overall amount of added time units needed. In such a way we can design secure but still time effective systems. This seems to be particularly important for various, frequently proprietary low energy protocols, employed by IoT systems, which include communications with variety of micro sensors with limited power supplies (see [Gar16,Hor17]).

5 Discussion and further work

We have presented the new security concept called initial process opacity and we have formalized it in the timed process algebra framework. We have proved some of its basic properties as well as properties of its variants, namely persistent and strong initial process opacity. Particularly we have investigated finite state systems and time sensitive observations. Moreover, by a careful choice of predicates defined by processes we can obtain properties which can be effectively checked. We can model security with respect to limited time length of an attack, with a limited number of attempts to perform an attack and so on. The presented approach allows us to exploit also process algebras enriched by operators expressing other "parameters" (space, distribution, networking architecture, processor or power consumption and so on). In this way also other types of attacks, which exploit information flow through various covert channels, can be described. Hence we could obtain security properties which have not only theoretical but also practical value, since many protocols, particularly low level protocols for IoT, could be described by means of process algebra formalism. Moreover, there are well developed techniques and software tools for process algebra's formal verification. We also plan to study security policies which assume intruders which are not only observers but can actively interact with the systems to be attacked.

References

[BKR04] Bryans J., M. Koutny and P. Ryan: Modelling non-deducibility using Petri Nets. Proc. of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models, 2004.

[BKMR06] Bryans J., M. Koutny, L. Mazare and P. Ryan: Opacity Generalised to Transition Systems. In Proceedings of the Formal Aspects in Security and Trust, LNCS 3866, Springer, Berlin, 2006.

[FGM00] Focardi, R., R. Gorrieri, and F. Martinelli: Information flow analysis in a discrete-time process algebra. Proc. 13^{th} Computer Security Foundation Workshop, IEEE Computer Society Press, 2000.

[Gar16] Garrido C., V. Lopez, T. Olivares and M. C. Ruiz: Architecture Proposal for Heterogeneous, BLE-Based Sensor and Actuator Networks for Easy Management of Smart Homes. 15th ACMIEEE International Conference on Information Processing in Sensor Networks (IPSN), 2016

[GM04] Gorrieri R. and F. Martinelli: A simple framework for real-time cryptographic protocol analysis with compositional proof rules. Science of Computer Programming, Volume 50, Issues 13, 2004.

[GM82] Goguen J.A. and J. Meseguer: Security Policies and Security Models. Proc. of IEEE Symposium on Security and Privacy, 1982.

[Gro90] Groote, J. F.: Transition Systems Specification with Negative Premises. Proc. of CONCUR'90, Springer Verlag, Berlin, LNCS 458, 1990.

[Gru15] Gruska D.P.: Process Opacity for Timed Process Algebra. In Perspectives of System Informatics, LNCS 8974, 2015.
Fundamenta Informaticae, vol. 102, Number 1, 2010.

[Gru07] Gruska D.P.: Observation Based System Security. Fundamenta Informaticae, vol. 79, Numbers 3-4, 2007.

[Hor17] Hortelano, D., T Olivares, M.C. Ruiz, M. Carmen, C. Garrido-Hidalgo and V. Lpez: rom Sensor Networks to Internet of Things. Bluetooth Low Energy, a Standard for This Evolution. Sensors, vol 17, 2017.

[JLF16] Jacob, R., J.-J.Lesage and J.-M. Faure: Overview of discrete event systems opacity: Models, validation, and quantification, Annual Reviews in Control Volume 41, 2016.